Private Information Retrieval Schemes with Product-Matrix MBR Codes

Julien Lavauzelle, Razane Tajeddine, Ragnar Freij-Hollanti, Camilla Hollanti

Private Information Retrieval Schemes with Product-Matrix MBR Codes

Julien Lavauzelle*†‡, Razane Tajeddine*¶, Ragnar Freij-Hollanti§, Camilla Hollanti§

† LIX, École Polytechnique, Inria & CNRS UMR 7161, University Paris-Saclay, Palaiseau, France
‡ Univ. Rennes, CNRS, IRMAR – UMR 6625, F-35000 Rennes, France
Email: julien.lavauzelle@univ-rennes1.fr
¶ Department of Computer Science, University of Helsinki, Helsinki, Finland
Emails: razane.tajeddine@helsinki.fi
§ Department of Mathematics and Systems Analysis, Aalto University School of Science, Espoo, Finland
Emails: {ragnar.freij, camilla.hollanti}@aalto.fi

***Abstract*—A private information retrieval (PIR) scheme allows a user to retrieve a file from a database without revealing any information on the file being requested. As of now, PIR schemes have been proposed for several kinds of storage systems, including replicated and MDS-coded systems. However, the problem of constructing PIR schemes on regenerating codes has been sparsely considered.**

A regenerating code is a storage code whose codewords are distributed among nodes, enabling efficient storage of files, as well as low-bandwidth retrieval of files and repair of nodes. Minimum-bandwidth regenerating (MBR) codes define a family of regenerating codes allowing a node repair with optimal bandwidth. Rashmi, Shah, and Kumar obtained a large family of MBR codes using the product-matrix (PM) construction.

In this work, a new PIR scheme over PM-MBR codes is designed. The inherent redundancy of the PM structure is used to reduce the download communication complexity of the scheme. A lower bound on the PIR capacity of MBR-coded PIR schemes is derived, showing an interesting storage space vs. PIR rate trade-off compared to existing PIR schemes with the same reconstruction capability. The present scheme also outperforms a recent PM-MBR PIR construction of Dorkson and Ng.

I. Introduction

Private information retrieval (PIR) allows a user to retrieve a file from a storage system without revealing which file she is interested in. The PIR problem was introduced by Chor, Goldreich, Kushilevitz and Sudan [1], [2]: given a tuple of files $\boldsymbol{X} = (\boldsymbol{X}^1, \ldots, \boldsymbol{X}^F)$ replicated over $n \geq 2$ servers, the goal is to retrieve a file $\boldsymbol{X}^f$ without revealing the index $f \in \{1, \ldots, F\}$ to the servers. While first constructions [3]–[5] mostly aimed at decreasing the overall communication complexity of the protocol, recent works only focused on the *PIR rate*, namely the ratio of downloaded information bits to the total number of downloaded bits. Sun and Jafar [6], [7] proved that the maximum PIR rate (called the PIR *capacity*) for F replicated files is $(1 - 1/n)/(1 - 1/n^F)$.

Many works also considered the PIR model where data is not replicated, but coded and distributed over multiple servers, see *e.g.* [8]–[17]. Symmetric PIR schemes were considered in [18], [19]. Moreover, the capacity of $[n, k]$-MDS-coded PIR schemes was found by Banawan and Ulukus [10]. This capacity converges exponentially fast to $1 - k/n$ for an unbounded number of files ($F \to \infty$) [9].

The present work will focus on the case of *regenerating codes* as storage codes. Regenerating codes are a class of codes dedicated to distributed storage, achieving the optimal trade-off between the bandwidth needed for a node repair and the amount of data each node needs to store. These codes were pioneered by Dimakis *et al.* [20] who notably produced a cut-set bound on the parameters of the codes. This bound materializes two interesting optimal settings: one for which the repair communication cost is minimized, called the minimum-bandwidth regenerating (MBR) point, and one for which the nodes store the least data, called the minimum-storage regenerating (MSR) point. Rashmi *et al.* [21] then proposed optimal constructions for these two specific settings, based on the so-called *product-matrix* (PM) framework. Many other works followed for the construction of MBR/MSR codes, including [22]–[26].

Main contributions. We propose a PIR scheme for the PM-MBR construction of Rashmi *et al.* [21]. We use symmetry and redundancy inherent to the PM framework as a way to decrease the number of symbols downloaded from the servers. As a consequence, we outperform the recent construction of PIR schemes over PM codes given by Dorkson and Ng in [27], [28], which represent the only existing works on PIR schemes for MBR codes.

We achieve a PIR rate strictly larger than $1 - k/n$, where n is the number of servers and k is the reconstruction parameter, *i.e.*, the minimum number of servers to be contacted in order to retrieve a file. This is achieved at the cost of an increased storage space requirement, allowing for efficient repair. Though we do not manage to compute the PIR capacity for MBR codes, our construction incurs a lower bound on it, and we can compare the PIR rate we obtained with achievable PIR rates over other storage systems with the same reconstruction

*: Both authors contributed equally to this manuscript.

parameter. Notice that the techniques presented in this paper also apply to the MSR setting, but reaches, in that case, a lower PIR rate than the PIR capacity of MDS and MSR storage systems [29], [30].

Organization. Section II introduces notation and definitions of private information retrieval, regenerating codes and product-matrix codes. The proposed PIR scheme is presented and analyzed in Section III, including an example with small parameters. Finally, a comparison of the PIR rate with some bounds and with PIR rates of other constructions is provided in Section IV.

II. Preliminaries

A. Notation and definitions

We let $\mathbb{F}_q$ denote the finite field with q elements. Given two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{F}_q^n$, their component-wise product is defined as $\boldsymbol{a} \star \boldsymbol{b} \coloneqq (a_1 b_1, \dots, a_n b_n) \in \mathbb{F}_q^n$. This operation easily extends to higher dimensional objects, such as matrices or arrays. Given a subset $\mathcal{I} \subset \{1, \dots, n\} =: [1, n]$, the tuple $\boldsymbol{a}_{|\mathcal{I}}$ is obtained by restricting $\boldsymbol{a}$ to coordinates in $\mathcal{I}$.

Let $1 \leq k \leq n$. The *Reed-Solomon code* of dimension k with distinct evaluation points $\boldsymbol{x} = (x_1, \dots, x_n)$, where $x_i \in \mathbb{F}_q$, is defined by

$$\mathrm{RS}_k(\boldsymbol{x}) \coloneqq \{(f(x_1), \dots, f(x_n)),\ f \in \mathbb{F}_q[X], \\ \deg f \leq k-1\} \subseteq \mathbb{F}_q^n\,.$$

It is well-known that $\mathrm{RS}_k(\boldsymbol{x})$ is maximum-distance separable (MDS), and that $\mathrm{RS}_j(\boldsymbol{x}) \subseteq \mathrm{RS}_k(\boldsymbol{x})$ for every $j \leq k$. Therefore there exists a basis $\Gamma = \{\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_k\}$ of $\mathrm{RS}_k(\boldsymbol{x})$ such that, for every $j \leq k$ and every subset $\mathcal{I} \subset \{1, \dots, n\}$ of size $|\mathcal{I}| \geq j$, the family $\Gamma^{(\mathcal{I},j)} \coloneqq \{(\boldsymbol{\gamma}_1)_{|\mathcal{I}}, \dots, (\boldsymbol{\gamma}_j)_{|\mathcal{I}}\}$ is a basis of $\mathrm{RS}_j(\boldsymbol{x}_{|\mathcal{I}}) \subseteq \mathbb{F}_q^{|\mathcal{I}|}$. For instance, one can take a degree-ordered monomial basis, explicitly given by $\boldsymbol{\gamma}_j \coloneqq (x_1^j, \dots, x_n^j) \in \mathbb{F}_q^n$.

The *Vandermonde matrix* with distinct basis elements $\boldsymbol{x} \in \mathbb{F}_q^n$ is the $n \times k$ matrix $\boldsymbol{\Psi} \in \mathbb{F}_q^{n \times k}$ such that $\Psi_{i,j} = x_i^j$ for $1 \leq i \leq n$ and $1 \leq j \leq k$. We know that $\boldsymbol{\Psi}$ generates the code $\mathrm{RS}_k(\boldsymbol{x})$ by columns. More precisely, these columns form the monomial basis we mentioned earlier. Notation is summarized in Table I.

B. Private information retrieval

Let $\mathcal{C}$ be a linear code of length n, and let us consider a storage system where n servers jointly store $F \geq 2$ codewords $\boldsymbol{C}^1, \dots, \boldsymbol{C}^F$ corresponding to files $\boldsymbol{X}^1, \dots, \boldsymbol{X}^F$. The i-th server stores $(C_i^1, \dots, C_i^F)$, *i.e.* the collection of i-th symbols of all files.

Assume now that a user wants to retrieve a specific file $\boldsymbol{X}^{f_0}$, for some $1 \leq f_0 \leq F$. In a private information retrieval (PIR) scheme, she sends *queries* $\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_n$ to servers, which compute *responses* $\boldsymbol{R}_1, \dots, \boldsymbol{R}_n$ accordingly. We say the scheme achieves information-theoretic PIR against non-colluding servers if the following holds:

$$\begin{aligned} &\text{Privacy:} && H(f_0 \mid \boldsymbol{Q}_i) = H(f_0), \quad i = 1, \dots, n. \\ &\text{Recovery:} && H(\boldsymbol{X}^{f_0} \mid \boldsymbol{R}_1, \dots, \boldsymbol{R}_n, \boldsymbol{Q}_1, \dots, \boldsymbol{Q}_n, f_0) = 0\,. \end{aligned}$$

TABLE I
Nomenclature

$\mathcal{C}$	Regenerating code
F	Number of files
n	Number of servers
k	Reconstruction parameter of the regenerating code
d	Repair parameter of the regenerating code
B	Number of symbols in a regenerating codeword
α	Storage capacity of a single server
β	Repair-bandwidth of a single server
$\boldsymbol{X} = (\boldsymbol{X}^1, \dots, \boldsymbol{X}^F)$	Set of files (database)
$\boldsymbol{X}^{f_0}$	Specific file requested by the user
$\boldsymbol{M}^f$	Redundant arrangement of the file $\boldsymbol{X}^f$ in a matrix (see PM framework)
$\boldsymbol{C}^f$	Regenerating codeword associated to $\boldsymbol{X}^f$, as stored on the DSS
$\boldsymbol{C}^{f,s}$	s-th stripe of the codeword $\boldsymbol{C}^f$
$\boldsymbol{C}^{f,s}[\cdot, j]$	j-th column of $\boldsymbol{C}^{f,s}$
$\boldsymbol{C}^{f,s}[i, \cdot]$	i-th row of $\boldsymbol{C}^{f,s}$ (stored by the i-th server)
$C^{f,s}[i, j]$	(i,j)-th symbol of stripe $\boldsymbol{C}^{f,s}$
$\boldsymbol{Q}_\ell$	ℓ-th query sent to servers
ρ	Rate of a PIR scheme
$H(\cdot)$	Entropy function

Here, $H(\cdot)$ denotes the entropy function. Concerning the recovery constraint, it is also desirable that the user is able to reconstruct $\boldsymbol{X}^{f_0}$ efficiently from $\boldsymbol{R}_1, \dots, \boldsymbol{R}_n$. We finally define the (download) PIR rate of a scheme by $\rho \coloneqq \frac{H(\boldsymbol{X}^{f_0})}{\sum_i H(\boldsymbol{R}_i)}$. The PIR capacity of a family of storage systems is the maximum achievable PIR rate.

C. Regenerating codes

Regenerating codes were introduced by Dimakis *et al.* in the context of distributed storage [20]. In an $(n, k, d, B, \alpha, \beta)$ regenerating code, a coded version of a file of size B is stored on n servers (or nodes), each storing α symbols, with the two following additional constraints. First, any external user must be able to retrieve any file by contacting any subset of k servers. Second, any failed server must be repairable by contacting any subset of $d \geq k$ servers and downloading β symbols from each, *i.e.*, $\gamma \coloneqq \beta d$ symbols in total. Parameters of regenerating codes are sometimes shortly denoted (n, k, d), but one should take care that d is *not* the minimum distance of the code, and k is *not* the dimension of the code.

Dimakis *et al.* [20] proved that any storage (erasure) code must satisfy the so-called *cut-set bound*

$$B \leq \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}\,, \tag{1}$$

and codes achieving this bound are called *regenerating codes*. The authors also proved that equality in (1) defines a trade-off between parameters α and $\gamma = \beta d$, which cannot be minimized simultaneously. Optimal codes minimizing $\gamma = \beta d$ reach the minimum-bandwidth regeneration (MBR) point, while those minimizing α attain the minimum-storage regeneration (MSR) point.

D. Product-Matrix MBR codes

In this work, we focus on MBR codes built by Rashmi *et al.* in [21] through the *product-matrix* (PM) framework. This

construction allows us to set $\beta = 1$, without loss of generality, since PM-MBR codes with $\beta \neq 1$ can be built from PM-MBR codes with $\beta = 1$ by *striping* files, see [21, Sect. I-C.]. File striping commonly refers to slicing files into subfiles; for instance a file of βN symbols can be sliced into β *stripes* (or subfiles) of N symbols each. In this setting we get the following constraints on the parameters:

$$\alpha = d \quad \text{and} \quad B = k(d-k) + \frac{k(k+1)}{2}.$$

The construction of Rashmi *et al.* [21] can be presented as follows. The symbols of a file $\boldsymbol{X} \in \mathbb{F}_q^B$ are arranged in a $d \times d$ matrix

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{S} & \boldsymbol{T} \\ \boldsymbol{T}^\top & \boldsymbol{0} \end{pmatrix} \tag{2}$$

where $\boldsymbol{S}$ is a $k \times k$ symmetric matrix containing $\frac{k(k+1)}{2}$ distinct file symbols, and $\boldsymbol{T}$ is a $k \times (d-k)$ matrix containing the remaining $k(d-k)$ file symbols. Notice that $\boldsymbol{M}$ is a symmetric matrix. Let now $\boldsymbol{\Psi}$ be an $n \times d$ Vandermonde matrix over $\mathbb{F}_q$. The codeword associated to file $\boldsymbol{X}$ is defined as

$$\boldsymbol{C} \coloneqq \boldsymbol{\Psi}\boldsymbol{M} \in \mathbb{F}_q^{n \times d}.$$

Let $\mathcal{C}$ be the set of all possible codewords $\boldsymbol{C}$ of the latter form. Notice that $\mathcal{C}$ is an $[nd, B]$ linear code over $\mathbb{F}_q$. In practice, the i-th row $\boldsymbol{C}[i, \cdot]$ of a codeword $\boldsymbol{C} \in \mathcal{C}$ is stored on server S_i, for $i = 1, \ldots, n$, and contains at most $\alpha = d$ information symbols. Let us now rewrite the example given by the authors in [21, Sec. IV.A.].

Example 1 (PM-MBR code)**.** *Consider the setting* $(n, k, d) = (6, 3, 4)$ *over* $\mathbb{F}_7$*. The original file contains* $B = 9$ *symbols. Let* $\boldsymbol{x} = (1, 2, 3, 4, 5, 6) \in \mathbb{F}_7^6$*. The generator (Vandermonde) matrix and the message matrix are then given as:*

$$\boldsymbol{\Psi} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 \\ 1 & 3 & 2 & 6 \\ 1 & 4 & 2 & 1 \\ 1 & 5 & 4 & 6 \\ 1 & 6 & 1 & 6 \end{pmatrix}, \quad \boldsymbol{M} = \begin{pmatrix} m_1 & m_2 & m_3 & m_7 \\ m_2 & m_4 & m_5 & m_8 \\ m_3 & m_5 & m_6 & m_9 \\ m_7 & m_8 & m_9 & 0 \end{pmatrix}.$$

III. A PIR scheme for product-matrix MBR codes

In this section, we consider a PM-MBR code $\mathcal{C}$ over $\mathbb{F}_q$ with parameters (n, k, d). Recall that $\mathcal{C}$ is a linear code over $\mathbb{F}_q$ of length nd and dimension $B = k(d-k) + \frac{k(k+1)}{2}$.

A. System setup

We consider a database $\boldsymbol{X}$ containing F files $\boldsymbol{X}^1, \ldots, \boldsymbol{X}^F$. For every $f \in \{1, \ldots, F\}$, the file $\boldsymbol{X}^f$ is subdivided into $S \geq 1$ *stripes* (or subdivisions), such that each stripe consists of $B = k(d-k) + \frac{k(k+1)}{2}$ information symbols.

For $s = 1, \ldots, S$, the s-th stripe of file $\boldsymbol{X}^f$ is then organized in a matrix $\boldsymbol{M}^{f,s}$ such that

$$\boldsymbol{M}^{f,s} = \left(M^{f,s}[i,j]\right)_{1 \leq i,j \leq d} \in \mathbb{F}_q^{d \times d}.$$

Following the PM framework, every stripe $\boldsymbol{M}^{f,s}$ must follow the form given in (2), and is encoded into a codeword $\boldsymbol{C}^{f,s} \in \mathcal{C}$. Explicitly, the j-th column vector of $\boldsymbol{C}^{f,s}$ is given by:

$$\boldsymbol{C}^{f,s}[\cdot, j] = \sum_{r=1}^{d} M^{f,s}[r,j]\boldsymbol{\gamma}_r,$$

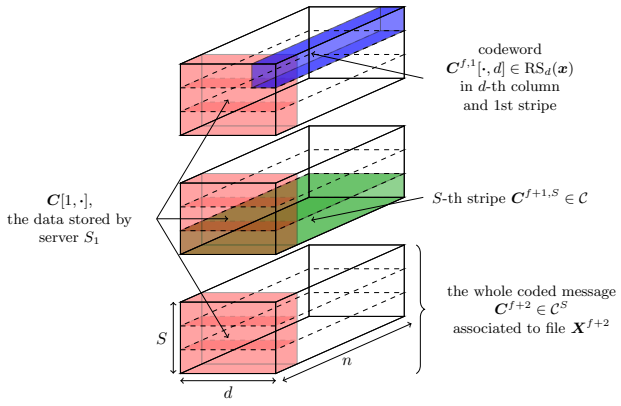


Fig. 1. An illustration of the arrangement of files, stripes and codewords in the storage system. A system of n servers stores coded files represented by $S \times d \times n$ cuboids (in the figure, only three of them are represented). Foreground (red) blocks represent data stored by the first server. The horizontal block (in green) in the middle cuboid represents a stripe, which lies in the regenerating code $\mathcal{C}$. Top right block (in blue) is a column of a stripe, and lies in an MDS code.

where we recall that $\Gamma = \{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_d\}$ denotes a suitable basis for sequences of Reed-Solomon codes (see Section II-A). Due to the shape of message matrices $\boldsymbol{M}^{f,s}$, one can also remark that $\boldsymbol{C}^{f,s}[\cdot, j] \in \mathrm{RS}_k(\boldsymbol{x})$ for every $j \geq k+1$. Illustration of the storage system is given in Figure 1.

B. Intuition

The idea behind the constructed PIR scheme is to take advantage of the symmetric property of matrices $\boldsymbol{M}^{f,s}$ as a way to reuse information, in order to decrease the download complexity of the scheme. We add that the servers are assumed not to collude. In this scheme, each file is divided into $S = n - k$ stripes. The user generates a set of k queries to the servers, similarly to the scheme in [13]. A query is defined as an $S \times F$-tuple of matrices that is sent by the user to retrieve information. Randomness is embedded in the queries as a way to hide the identity of requested file, in a similar manner to one-time padding. Naturally, if privacy were not a concern, a retrieval query for file $\boldsymbol{X}^{f_0}$ would be the tuple of matrices with zero-matrices everywhere, except for index $f = f_0$.

Queries are then sent to servers which project them on their stored data in the following way. For the last $d-k$ columns, since each of these columns stores file stripes encoded using an $[n, k]$ MDS code, servers are asked to project *all* the queries on the data they hold, similarly to [13]. For each of the other columns, stripes contain information already retrieved from the previously used columns, due to the nature of the product-matrix construction. Thus, from server S_k down to server S_1, servers are asked to project on their stored data a smaller subset of the initial set of queries. This still enables the user to reconstruct the requested file, due to the fact that she had peeled off some randomness and information symbols from previous columns. Moreover, it allows her to run locally a more efficient PIR scheme, since the punctured queries apply to an MDS code with lower information rate. More details are given in the upcoming sections.

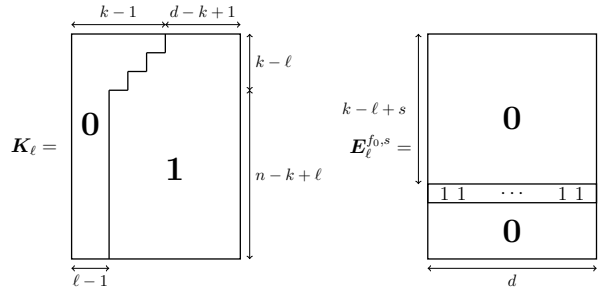


Fig. 2. Representation of the mask matrix $\boldsymbol{K}_\ell$ (on the left) and the retrieval pattern $\boldsymbol{E}_\ell^{f_0,s}$ (on the right).

C. The PIR scheme

In this section, we describe the PIR scheme explicitly. Let us assume that the user wants to retrieve a file $\boldsymbol{X}^{f_0}$, for some $1 \le f_0 \le F$. We fix the number of stripes to $S = n-k$, and we consider a k-tuple of queries $\boldsymbol{Q} = (\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_k)$, such that for each $\ell \in \{1, \dots, k\}$, query $\boldsymbol{Q}_\ell$ is an $F \times S$ tuple of matrices $\boldsymbol{Q}_\ell^{f,s} \in \mathbb{F}_q^{n \times d}$.

For each $1 \le i \le n$, the collection of i-th rows of matrices $\boldsymbol{Q}_\ell^{f,s} \in \mathbb{F}_q^{n \times d}$ are sent to server S_i. The response $\boldsymbol{R}_\ell \in \mathbb{F}_q^{n \times d}$ is then defined as:

$$\boldsymbol{R}_\ell \coloneqq \sum_{f=1}^{F} \sum_{s=1}^{n-k} \boldsymbol{Q}_\ell^{f,s} \star \boldsymbol{C}^{f,s}\,.$$

Throughout the paper, we will use $\sum_{f,s}$ instead of $\sum_{f=1}^{F}\sum_{s=1}^{n-k}$ as an abuse of notation.

Generation of $\boldsymbol{Q}$. Let us denote by $\mathbf{1} \in \mathbb{F}_q^{n \times d}$ the all-one matrix. For $\ell \in \{1, \dots, k\}$ we define the *mask* $\boldsymbol{K}_\ell \in \mathbb{F}_q^{n \times d}$ as follows:

$$K_\ell[i,j] = \begin{cases} 0 & \text{if } j < \ell \text{ or } i + j \le k-1, \\ 1 & \text{otherwise.} \end{cases}$$

Let us fix $f_0 \in \{1, \dots, F\}$ the index of the required file. We define the *retrieval pattern* $\boldsymbol{E}_\ell^{f,s} \in \mathbb{F}_q^{n \times d}$ by:

$$\boldsymbol{E}_\ell^{f,s} = \mathbf{0} \quad \text{if } f \ne f_0$$

and for every j,

$$E_\ell^{f_0,s}[i,j] = \begin{cases} 1 & \text{if } i = k+s-\ell+1, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that i is well-defined since s ranges in $\{1, \dots, n-k\}$ and ℓ ranges in $\{1, \dots, k\}$. Figure 2 proposes an illustration of matrices $\boldsymbol{K}_\ell^{f,s}$ and $\boldsymbol{E}_\ell^{f_0,s}$.

Finally, queries $\boldsymbol{Q}$ are defined as follows. For every ℓ, s, f, symbols $\lambda_\ell^{f,s} \in \mathbb{F}_q$ are picked uniformly at random and independently. Then, we set:

$$\boldsymbol{Q}_\ell^{f,s} \coloneqq (\lambda_\ell^{f,s}\mathbf{1} + \boldsymbol{E}_\ell^{f,s}) \star \boldsymbol{K}_\ell\,.$$

Server responses to queries. Due to the shape of mask matrices, one can reduce the download communication complexity of the scheme. Indeed, the i-th server must produce only response symbols corresponding to ones in row i of masks $\boldsymbol{K}_1, \dots, \boldsymbol{K}_k$. For $i \le k$ we get

$$k(d-k+1) + \sum_{\ell=1}^{i-1}(k-\ell) = k(d-k+i) - \frac{i(i-1)}{2}$$

ones in i-th rows. Similarly, for $i > k$ the i-th server must compute only $dk - k(k-1)/2$ response symbols.

Recovery of $\boldsymbol{X}^{f_0}$. The recovery is run column-wise, from column d down to column 1. For each step j, $1 \le j \le d$, the goal is to retrieve column vectors $\boldsymbol{M}^{f_0,s}[\cdot, j]$ along with some random vectors.

- *For $j \in \{k+1, \dots, d\}$.* A precise description of the recovery algorithm is given in the proof of Lemma 1. In short, it consists of running, *independently on each column*, the reconstruction of the PIR scheme over an MDS code described in [13]. Indeed, the part of the database corresponding to the j-th column — namely $\boldsymbol{C}[\cdot, j]$ — can be viewed as an $[n,k]$ MDS-coded database. This procedure allows the user to recover striped columns $\boldsymbol{M}^{f_0,s}[\cdot, j]$ of the desired file, but she can also collect random vectors $\sum_{f,s} \lambda_\ell^{f,s} \boldsymbol{M}^{f,s}[\cdot, j] \in \mathbb{F}_q^n$, for all $1 \le \ell \le k$.
- *For $j \in \{1, \dots, k\}$.* At step j, one can assume that for every $j' \ge j+1$, the user has already collected
  - $\boldsymbol{M}^{f_0,s}[\cdot, j']$ for every $s \in \{1, \dots, S\}$, and
  - random vectors $\sum_{f,s} \lambda_\ell^{f,s} \boldsymbol{M}^{f,s}[\cdot, j'] \in \mathbb{F}_q^n$ for every $1 \le \ell \le k$.

  Since matrices $\boldsymbol{M}^{f,s}$ are symmetric, the user knows $\sum_{f,s} \lambda_\ell^{f,s} M^{f,s}[i,j]$ for every $j+1 \le i \le d$ and every $1 \le \ell \le j$. Informally, this knowledge allows her to run an $[n-k+j, j]$-MDS-PIR scheme on the data stored on the j-th column. The retrieval process is described precisely in the proof of Lemma 2. It ensures that the user can retrieve $\boldsymbol{M}^{f_0,s}[\cdot, j]$ for all s, and random vectors $\sum_{f,s} \lambda_\ell^{f,s} \boldsymbol{M}^{f,s}[\cdot, j]$ for every $1 \le \ell \le j$.

We start by giving a simple example before diving into technical proofs.

Example 2. *We use the $(6,3,4)$ PM-MBR regenerating code described in Example 1. For this purpose, the files are divided into $S = n-k = 3$ stripes, and the user sends $k = 3$ query vectors:*

For $f = f_0$, the first ($\ell = 1$) query vector $(\boldsymbol{Q}_1^{f_0,1}, \boldsymbol{Q}_1^{f_0,2}, \boldsymbol{Q}_1^{f_0,3})$ consists of matrices:

$$\begin{pmatrix} 0 & 0 & u_1 & u_1 \\ 0 & u_1 & u_1 & u_1 \\ u_1 & u_1 & u_1 & u_1 \\ u_1+1 & u_1+1 & u_1+1 & u_1+1 \\ u_1 & u_1 & u_1 & u_1 \\ u_1 & u_1 & u_1 & u_1 \end{pmatrix},$$

$$\begin{pmatrix} 0 & 0 & v_1 & v_1 \\ 0 & v_1 & v_1 & v_1 \\ v_1 & v_1 & v_1 & v_1 \\ v_1 & v_1 & v_1 & v_1 \\ v_1+1 & v_1+1 & v_1+1 & v_1+1 \\ v_1 & v_1 & v_1 & v_1 \end{pmatrix},$$

and

$$\begin{pmatrix} 0 & 0 & w_1 & w_1 \\ 0 & w_1 & w_1 & w_1 \\ w_1 & w_1 & w_1 & w_1 \\ w_1 & w_1 & w_1 & w_1 \\ w_1 & w_1 & w_1 & w_1 \\ w_1+1 & w_1+1 & w_1+1 & w_1+1 \end{pmatrix},$$

TABLE II
RESPONSES TO THE FIRST QUERY IN EXAMPLE 2

	Response column $\boldsymbol{R}_1[\cdot,4]$
Server S_1	$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_1^{f,s}C^{f,s}[1,4]$
Server S_2	$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_1^{f,s}C^{f,s}[2,4]$
Server S_3	$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_1^{f,s}C^{f,s}[3,4]$
Server S_4	$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_1^{f,s}C^{f,s}[4,4]+C^{f_0,1}[4,4]$
Server S_5	$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_1^{f,s}C^{f,s}[5,4]+C^{f_0,2}[5,4]$
Server S_6	$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_1^{f,s}C^{f,s}[6,4]+C^{f_0,3}[6,4]$

where $(u_1,v_1,w_1)=(\lambda_1^{f_0,1},\lambda_1^{f_0,2},\lambda_1^{f_0,3})$. *Queries for* $f\neq f_0$ *can be derived from previous matrices by removing the "+1" pattern. Similarly, the second* $(\ell=2)$ *query vector* $(\boldsymbol{Q}_2^{f_0,1},\boldsymbol{Q}_2^{f_0,2},\boldsymbol{Q}_2^{f_0,3})$ *is:*

$$\begin{pmatrix}0&0&u_2&u_2\\0&u_2&u_2&u_2\\0&u_2+1&u_2+1&u_2+1\\0&u_2&u_2&u_2\\0&u_2&u_2&u_2\\0&u_2&u_2&u_2\end{pmatrix},\begin{pmatrix}0&0&v_2&v_2\\0&v_2&v_2&v_2\\0&v_2&v_2&v_2\\0&v_2+1&v_2+1&v_2+1\\0&v_2&v_2&v_2\\0&v_2&v_2&v_2\end{pmatrix},$$

and

$$\begin{pmatrix}0&0&w_2&w_2\\0&w_2&w_2&w_2\\0&w_2&w_2&w_2\\0&w_2&w_2&w_2\\0&w_2+1&w_2+1&w_2+1\\0&w_2&w_2&w_2\end{pmatrix},$$

where $(u_2,v_2,w_2)=(\lambda_2^{f_0,1},\lambda_2^{f_0,2},\lambda_2^{f_0,3})$, *and the third* $(\ell=3)$ *query vector* $(\boldsymbol{Q}_3^{f_0,1},\boldsymbol{Q}_3^{f_0,2},\boldsymbol{Q}_3^{f_0,3})$ *is:*

$$\begin{pmatrix}0&0&u_3&u_3\\0&0&u_3+1&u_3+1\\0&0&u_3&u_3\\0&0&u_3&u_3\\0&0&u_3&u_3\\0&0&u_3&u_3\end{pmatrix},\begin{pmatrix}0&0&v_3&v_3\\0&0&v_3&v_3\\0&0&v_3+1&v_3+1\\0&0&v_3&v_3\\0&0&v_3&v_3\\0&0&v_3&v_3\end{pmatrix},$$

and

$$\begin{pmatrix}0&0&w_3&w_3\\0&0&w_3&w_3\\0&0&w_3&w_3\\0&0&w_3+1&w_3+1\\0&0&w_3&w_3\\0&0&w_3&w_3\end{pmatrix},$$

where $(u_3,v_3,w_3)=(\lambda_3^{f_0,1},\lambda_3^{f_0,2},\lambda_3^{f_0,3})$.

• **Decodability:** *Decoding is done iteratively from column* $d=4$ *to column* 1.

∗ 4-th column. *Responses to the first query are listed in Table 2.*

The column vector $\boldsymbol{R}_1[\cdot,4]$ *represented in Table 2 can be viewed as a random linear combination of codewords* $\boldsymbol{C}^{f,s}[\cdot,4]\in\mathrm{RS}_3(\boldsymbol{x})\subseteq\mathbb{F}_q^6$, *corrupted with* 3 *erasures. Precisely, let us denote the random message*

$$\boldsymbol{m}\coloneqq\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_1^{f,s}\boldsymbol{M}^{f,s}[\cdot,4]\in\mathbb{F}_q^4,$$

the random codeword

$$\boldsymbol{a}\coloneqq\boldsymbol{\Psi}\boldsymbol{m}=\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_1^{f,s}\boldsymbol{C}^{f,s}[\cdot,4]\in\mathrm{RS}_3(\boldsymbol{x}),$$

and the erasure vector

$$\boldsymbol{b}\coloneqq(0,0,0,C^{f_0,1}[4,4],C^{f_0,2}[5,4],C^{f_0,3}[6,4])^\top.$$

Also, let $\boldsymbol{\Psi}_{|[*,1:3]}$ *be the submatrix of* $\boldsymbol{\Psi}$ *formed by its three first columns. Note that we have* $\boldsymbol{R}_1[\cdot,4]=\boldsymbol{a}+\boldsymbol{b}$ *and* $\boldsymbol{R}_1[\cdot,4]_{|[1:3]}=\boldsymbol{a}_{|[1:3]}=\boldsymbol{\Psi}_{|[*,1:3]}\boldsymbol{m}_{|[1:3]}$, *since* $m_4=b_1=b_2=b_3=0$.

The submatrix $\boldsymbol{\Psi}_{|[*,1:3]}$ *is invertible, hence one can recover* $\boldsymbol{m}_{|[1:3]}$. *Then, one can successively get* $\boldsymbol{m}$ *since* $m_4=0$, $\boldsymbol{a}=\boldsymbol{\Psi}\boldsymbol{m}$, *and* $\boldsymbol{b}=\boldsymbol{R}_1[\cdot,4]-\boldsymbol{a}$. *Notice that the recovery of* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *succeeds because* $\mathrm{RS}_3(\boldsymbol{x})$ *has dimension* 3 *and is MDS, and the erasure vector* $\boldsymbol{b}$ *has weight at most* $6-3=3$.

For $\boldsymbol{R}_2[\cdot,4]$, *the reasoning is similar, except that the erasure vector is now*

$$\boldsymbol{b}'\coloneqq(0,0,C^{f_0,1}[3,4],C^{f_0,2}[4,4],C^{f_0,3}[5,4],0)^\top.$$

Hence its support is $\{3,4,5\}$, *so we need to use* $\boldsymbol{\Psi}_{|[*,\{1,2,6\}]}$ *instead of* $\boldsymbol{\Psi}_{|[*,1:3]}$. *Then, the user can also extract*

$$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_2^{f,s}\boldsymbol{C}^{f,s}[\cdot,4]\in\mathrm{RS}_3(\boldsymbol{x})$$

by solving a linear system, and gets $\boldsymbol{b}'$.

Following the same reasoning, from $\boldsymbol{R}_3[\cdot,4]$ *the user gets*

$$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_3^{f,s}\boldsymbol{C}^{f,s}[\cdot,4]\in\mathrm{RS}_3(\boldsymbol{x})$$

and

$$(0,C^{f_0,2}[2,4],C^{f_0,3}[3,4],C^{f_0,1}[4,4],0,0)^\top.$$

Now notice that for each $s\in\{1,2,3\}$, *the user has recovered* 3 *symbols of* $\boldsymbol{C}^{f_0,s}[\cdot,4]$ *(the coordinates of these symbols depend on* s*). Since* $\boldsymbol{C}^{f_0,s}[\cdot,4]$ *is the encoded version of* $\boldsymbol{M}^{f_0,s}[\cdot,4]$ *in* $\mathrm{RS}_3(\boldsymbol{x})$, *an MDS code of dimension* 3, *the user can recover the* 3 *message symbols* $M^{f_0,s}[1,4],M^{f_0,s}[2,4],M^{f_0,s}[3,4]$.

Finally, the user also retrieves random symbols

$$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_\ell^{f,s}M^{f,s}[i,4]$$

for every $\ell\in\{1,2,3\}$ *and every* $i\in\{1,2,3\}$.

The retrieval described above achieves a PIR rate of $3/6$.

∗ 3-rd column. *Here the (column) storage code is a* $[6,4]$ *MDS code since the third column of message matrices* $\boldsymbol{M}^{f,s}$ *has* 4 *non-zero entries. However, conditioned on the information retrieved from column 4, it can be regarded as a* $[6,3]$ *MDS storage code. Namely, recall that* $M^{f,s}[3,4]=M^{f,s}[4,3]$ *for every* f,s. *Hence, from the previous decoding step, the user knows* $M^{f_0,s}[4,3]$ *for every* $s\in\{1,2,3\}$, *as well as random symbols*

$$\sum_{f=1}^{F}\sum_{s=1}^{3}\lambda_\ell^{f,s}M^{f,s}[4,3]$$

for every $\ell\in\{1,2,3\}$.

Now, the user can write

$$\boldsymbol{R}_\ell[\cdot,3] = \boldsymbol{\Psi}_{[*,1:3]}\Big(\sum_{f,s}\lambda_\ell^{f,s}\boldsymbol{M}^{f,s}[\cdot,3]\Big) + \Big(\sum_{f,s}\lambda_\ell^{f,s}M^{f,s}[4,3]\Big)\boldsymbol{\Psi}_{[*,4:4]} + \boldsymbol{b}_\ell,$$

where $\boldsymbol{\Psi}_{[*,a:b]}$ denotes the sub-Vandermonde matrix with columns from a to b, and $\boldsymbol{b}_\ell$ is an erasure vector containing some desired information. For instance, we have $\boldsymbol{b}_2 = (0,0,C^{f_0,1}[3,3],C^{f_0,1}[4,3],C^{f_0,1}[5,3],0)^\top$. By subtracting $(\sum_{f,s}\lambda_\ell^{f,s}M^{f,s}[4,3])\boldsymbol{\Psi}_{[*,4:4]}$ from $\boldsymbol{R}_\ell[\cdot,3]$, the user can thus recover $\boldsymbol{b}_\ell$ as an erasure corrupting the $[6,3]$ MDS codeword $\boldsymbol{\Psi}_{[*,1:3]}(\sum_{f,s}\lambda_\ell^{f,s}\boldsymbol{M}^{f,s}[\cdot,3])$. She can then deduce $\boldsymbol{M}^{f_0,s}[\cdot,3]$ for every $s\in\{1,2,3\}$, along with randomness $\sum_{f,s}\lambda_\ell^{f,s}\boldsymbol{M}^{f,s}[\cdot,3]$ for every $\ell\in\{1,2,3\}$, with a rate $3/6$.

Notice that the decoding could be performed precisely because the user is able to use some information previously collected during the decoding of the 4-th column.

$*$ 2-nd column. *For this column, server S_1 does not send any response. Using the same techniques as for 3rd column, the user can represent responses $\boldsymbol{R}_\ell[\cdot,3]$ as a sum of 3 components: (i) previously collected information*

$$\sum_{f,s}\lambda_\ell^{f,s}\Big(M^{f,s}[4,2]\boldsymbol{\Psi}_{[1:5,4:4]} + M^{f,s}[3,2]\boldsymbol{\Psi}_{[1:5,3:3]}\Big)$$

since $M^{f,s}[4,2]=M^{f,s}[2,4]$ and $M^{f,s}[3,2]=M^{f,s}[2,3]$ are known, (ii) a codeword from a $[5,2]$ MDS code

$$\sum_{f,s}\lambda_\ell^{f,s}\Big(M^{f,s}[1,2]\boldsymbol{\Psi}_{[1:5,1:1]} + M^{f,s}[2,2]\boldsymbol{\Psi}_{[1:5,2:2]}\Big),$$

and (iii) an erasure containing symbols of the requested file. Similarly to 4-th and 3-rd columns, by solving linear systems, the user retrieves requested data $\boldsymbol{M}^{f_0,s}[\cdot,3]$ for every s, and random symbols $\sum_{f,s}\lambda_\ell^{f,s}\boldsymbol{M}^{f,s}[\cdot,3]$ for every ℓ. Here the retrieval rate is $3/5$.

$*$ 1-st column. *Servers 1 and 2 do not send any response. Analogously to steps 2 and 3, conditioned on previously retrieved information, the restriction of the storage code to servers 3 to 6 is equivalent to a $[4,1]$ MDS code. This allows the user to decode the last part of the file ($M^{f_0,s}[1,1]$ for every stripe s) at rate $3/4$.*

Finally, the PIR rate is $\rho=\frac{3+6+9+9}{4+10+18+18}=\frac{27}{50}=0.54$. This rate is larger than $1-\frac{k}{n}=1-\frac{3}{6}=\frac{1}{2}=0.5$ which is the asymptotic capacity of scalar MDS-coded PIR schemes, but less than $1-\frac{B}{nd}=1-\frac{9}{6\times4}=\frac{5}{8}=0.625$, the asymptotic capacity of scalar MDS-coded $[nd,B]$-coded PIR schemes.

- **Privacy:** *Each query received by a server is of the form*

$$(0,\dots,0,z,\dots z)$$

where $z\in\mathbb{F}_q$ is uniformly random, and the number of zeroes only depends on the server and the stripe index. Therefore, the scheme is private.

$\boldsymbol{B}_1[\cdot,j]$	$\boldsymbol{B}_\ell[\cdot,j]$	$\boldsymbol{B}_k[\cdot,j]$
0	0	0
$\vdots$	$\vdots$	$C^{f_0,1}[2,j]$
	$\vdots$	$\vdots$
	0	$C^{f_0,n-k}[n-k+1,j]$
$\vdots$	$C^{f_0,1}[k+2-\ell,j]$	0
0	$\vdots$	$\vdots$
$C^{f_0,1}[k+1,j]$	$C^{f_0,n-k}[n+1-\ell,j]$	
$\vdots$	0	
$C^{f_0,n-k}[n,j]$	0	0

Fig. 3. Representation of vectors $\boldsymbol{B}_\ell[\cdot,j]$ introduced in Lemma 1, for some $j\in\{k+1,\dots,d\}$.

D. Analysis

In this section, we prove the correctness of the PIR scheme introduced in the previous section.

Lemma 1. *Let $j\in\{k+1,\dots,d\}$ be a column index. Then, conditioned on $(\boldsymbol{R}_1[\cdot,j],\dots,\boldsymbol{R}_k[\cdot,j])$, the following are determined:*

- *the j-th column $\boldsymbol{M}^{f_0,s}[\cdot,j]$ of the desired file, for every stripe index $s\in\{1,\dots,n-k\}$;*
- *random vectors $\sum_{f,s}\lambda_\ell^{f,s}\boldsymbol{M}^{f,s}[\cdot,j]\in\mathbb{F}_q^n$ for every $1\le\ell\le k$.*

Proof. Let us fix $1\le\ell\le k$. Since $\boldsymbol{K}_\ell[\cdot,j]$ is the all-one vector, we have

$$\boldsymbol{R}_\ell[\cdot,j] = \sum_{s,f}\lambda_\ell^{f,s}\boldsymbol{C}^{f,s}[\cdot,j] + \sum_s \boldsymbol{E}_\ell^{f_0,s}[\cdot,j]\star\boldsymbol{C}^{f_0,s}[\cdot,j]\,.$$

We can now define

$$\boldsymbol{B}_\ell[\cdot,j] \coloneqq \sum_s \boldsymbol{E}_\ell^{f_0,s}[\cdot,j]\star\boldsymbol{C}^{f_0,s}[\cdot,j]\in\mathbb{F}_q^n\,,$$

and we see (illustrated in Figure 3) that

$$B_\ell[i,j]=\begin{cases} C^{f_0,s'}[i,j] & \text{if } i=k+s'-\ell+1,\\ 0 & \text{otherwise.}\end{cases} \tag{3}$$

In particular, vector $\boldsymbol{B}_\ell[\cdot,j]$ is supported on $\mathcal{J}_\ell\coloneqq\{k+2-\ell,\dots,n+1-\ell\}$, and therefore has weight at most $n-k$.

Now, let us introduce

$$\boldsymbol{A}_\ell[\cdot,j]\coloneqq\sum_{s,f}\lambda_\ell^{f,s}\boldsymbol{C}^{f,s}[\cdot,j]\in\mathrm{RS}_k(\boldsymbol{x})\,.$$

Since $R_\ell[i,j]=A_\ell[i,j]$ for $i\notin\mathcal{J}_\ell$, the user knows k symbols of $\boldsymbol{A}_\ell[\cdot,j]$ indexed by an information set $[1,n]\backslash\mathcal{J}_\ell$ for $\mathrm{RS}_k(\boldsymbol{x})$. Hence she can recover $\boldsymbol{A}_\ell[\cdot,j]$ entirely by solving a linear system of rank k. The recovery of $\boldsymbol{B}_\ell[\cdot,j]=\boldsymbol{R}_\ell[\cdot,j]-\boldsymbol{A}_\ell[\cdot,j]$ easily follows.

Let us now recall that $\boldsymbol{C}^{f,s}[\cdot,j]\in\mathrm{RS}_k(\boldsymbol{x})$ can be written as $\sum_{r=1}^k M^{f,s}[r,j]\boldsymbol{\gamma}_r$. Therefore, expressing

$$\boldsymbol{A}_\ell[\cdot,j]=\sum_{r=1}^d\left(\sum_{s,f}\lambda_\ell^{f,s}M^{f,s}[r,j]\right)\boldsymbol{\gamma}_r$$

in the basis $\{\boldsymbol{\gamma}_1,\dots,\boldsymbol{\gamma}_d\}\subset\mathbb{F}_q^n$ of the Reed-Solomon code $\mathrm{RS}_d(\boldsymbol{x})\supseteq\mathrm{RS}_k(\boldsymbol{x})$ allows us to retrieve every

scalar $\sum_{s,f} \lambda_\ell^{s,f} M^{f,s}[r,j]$. Finally, Equation (3) shows that the knowledge of $\boldsymbol{B}_1[\cdot,j],\dots,\boldsymbol{B}_k[\cdot,j]$ corresponds to the knowledge of k symbols of each of the $(n-k)$ stripes $\boldsymbol{C}^{f_0,1}[\cdot,j],\dots,\boldsymbol{C}^{f_0,n-k}[\cdot,j]$. Since each stripe lies in the MDS code $\mathrm{RS}_k(\boldsymbol{x})$, it leads to the recovery of all message stripes $\boldsymbol{M}^{f_0,1}[\cdot,j],\dots \boldsymbol{M}^{f_0,n-k}[\cdot,j]$. □

Lemma 2. *Let $j \in \{1,\dots,k\}$. For convenience we set $n' \coloneqq n-k+j$ and $\mathcal{I} \coloneqq \{k-j+1,\dots,n\}$. For every $1 \le \ell \le j$, we define*

$$\boldsymbol{R}_\ell[\mathcal{I},j] \coloneqq \boldsymbol{R}_\ell[\cdot,j]_{|\mathcal{I}} = (R_\ell[k-j+1,j],\dots,R_\ell[n,j]) \in \mathbb{F}_q^{n'}.$$

Then, conditioned on $(\boldsymbol{R}_1[\mathcal{I},j],\dots,\boldsymbol{R}_j[\mathcal{I},j])$ and on

$$\sum_{f,s} \lambda_\ell^{f,s} M^{f,s}[r,j], \quad \text{for all } \; j+1 \le r \le d, \quad 1 \le \ell \le j, \tag{4}$$

the following are determined:

- *column $\boldsymbol{M}^{f_0,s}[\cdot,j]$ of the desired file, for every stripe index $s \in \{1,\dots,n-k\}$;*
- *random elements $\sum_{f,s} \lambda_\ell^{f,s} M^{f,s}[r,j] \in \mathbb{F}_q^{n'}$ for all $1 \le r,\ell \le j$.*

Proof. Let us fix $\ell \in \{1,\dots,j\}$ and define $\boldsymbol{x}' \coloneqq \boldsymbol{x}_{|\mathcal{I}} = (x_{k-j+1},\dots,x_n)$ and $\boldsymbol{\gamma}'_r = (\gamma_r)_{|\mathcal{I}}$ for every $r \in \{1,\dots,d\}$. We decompose $\boldsymbol{R}_\ell[\mathcal{I},j] = \boldsymbol{A}_\ell[\mathcal{I},j] + \boldsymbol{B}_\ell[\mathcal{I},j]$ where

$$\boldsymbol{A}_\ell[\mathcal{I},j] \coloneqq \sum_{f,s} \lambda_\ell^{f,s} \boldsymbol{C}^{f,s}[\mathcal{I},j] \in \mathrm{RS}_d(\boldsymbol{x}')$$

and

$$\boldsymbol{B}_\ell[\mathcal{I},j] \coloneqq \sum_{s} \boldsymbol{E}_\ell^{f_0,s}[\mathcal{I},j] \star \boldsymbol{C}^{f_0,s}[\mathcal{I},j] \in \mathbb{F}_q^{n'}$$

are defined similarly to Lemma 1. One can rewrite $\boldsymbol{A}_\ell[\mathcal{I},j]$ as follows:

$$\begin{aligned}\boldsymbol{A}_\ell[\mathcal{I},j] &= \sum_{f,s} \lambda_\ell^{f,s} \left(\sum_{r=1}^{d} M^{f,s}[r,j]\, \boldsymbol{\gamma}'_r \right) \\ &= \sum_{r=1}^{j} \Big(\sum_{f,s} \lambda_\ell^{f,s} M^{f,s}[r,j] \Big) \boldsymbol{\gamma}'_r \\ &\qquad + \sum_{r=j+1}^{d} \Big(\sum_{f,s} \lambda_\ell^{f,s} M^{f,s}[r,j] \Big) \boldsymbol{\gamma}'_r .\end{aligned}$$

Therefore, using (4) one can deduce

$$\boldsymbol{A}'_\ell[\mathcal{I},j] \coloneqq \sum_{r=j+1}^{d} \Big(\sum_{f,s} \lambda_\ell^{f,s} M^{f,s}[r,j] \Big) \boldsymbol{\gamma}'_r \,.$$

Le us now define

$$\begin{aligned}\boldsymbol{R}'_\ell[\mathcal{I},j] &\coloneqq \boldsymbol{R}_\ell[\mathcal{I},j] - \boldsymbol{A}'_\ell[\mathcal{I},j] \\ &= (\boldsymbol{A}_\ell[\mathcal{I},j] - \boldsymbol{A}'_\ell[\mathcal{I},j]) + \boldsymbol{B}_\ell[\mathcal{I},j] \,.\end{aligned}$$

The set $\{\boldsymbol{\gamma}'_1,\dots,\boldsymbol{\gamma}'_j\}$ forms a basis of $\mathrm{RS}_j(\boldsymbol{x}')$, hence $\boldsymbol{A}''_\ell[\mathcal{I},j] \coloneqq \boldsymbol{A}_\ell[\mathcal{I},j] - \boldsymbol{A}'_\ell[\mathcal{I},j] \in \mathrm{RS}_j(\boldsymbol{x}')$. Also remark that the vector $\boldsymbol{B}_\ell[\mathcal{I},j] \in \mathbb{F}_q^{n'}$ is supported on $\mathcal{J}_\ell \coloneqq \{k+2-\ell,\dots,n+1-\ell\} \subset \mathcal{I}$. Since $\mathcal{I} \setminus \mathcal{J}_\ell$ has size j, it is an information set for the MDS code $\mathrm{RS}_j(\boldsymbol{x}')$, and one can thus recover $\boldsymbol{A}''_\ell[\mathcal{I},j]$ and $\boldsymbol{B}_\ell[\mathcal{I},j]$ from $\boldsymbol{R}''_\ell[\mathcal{I},j]$.

Similarly to Lemma 1, one can easily see that for each stripe s, codewords symbols $\boldsymbol{C}^{f_0,s}[\mathcal{I},j]$ and, thus, message symbols $\boldsymbol{M}^{f_0,s}[\mathcal{I},j]$ can be obtained from $\boldsymbol{B}_1[\mathcal{I},j],\dots,\boldsymbol{B}_j[\mathcal{I},j]$. Notice that this determines $\boldsymbol{M}^{f_0,s}[\cdot,j]$ entirely thanks to (4). Finally, $\boldsymbol{A}'_\ell[\mathcal{I},j]$ and $\boldsymbol{A}''_\ell[\mathcal{I},j]$ allow to reconstruct $\boldsymbol{A}_\ell[\mathcal{I},j]$. Similarly to the proof of Lemma 1, the basis $\{\boldsymbol{\gamma}'_1,\dots,\boldsymbol{\gamma}'_j\}$ of $\mathrm{RS}_j(\boldsymbol{x}')$ leads to the recovery of random elements $\sum_{f,s} \lambda_\ell^{f,s} M^{f,s}[r,j] \in \mathbb{F}_q$ for every $1 \le r,\ell \le j$. □

Theorem 1. *The scheme proposed in Section III-C is secure against non-colluding servers. Its PIR rate is:*

$$\rho = \frac{3(n-k)(2d-k+1)}{6dn-3nk+3n-k^2+1} \,.$$

Proof. Lemma 1 and Lemma 2 ensure that the user retrieves the correct file $\boldsymbol{X}^{f_0}$ as long as the servers $S_1,\dots,S_n$ follow the protocol described in Section III-C. Since the servers are assumed not to collude, the only way a server S_i can learn information about the identity f_0 of the required file, is from its own set of queries $\boldsymbol{Q}[i,\cdot] \coloneqq (\boldsymbol{Q}_\ell^{f,s}[i,\cdot])_{\ell,f,s}$. For fixed i, we have

$$\boldsymbol{Q}_\ell^{f,s}[i,\cdot] = (0,\dots,0,z,\dots,z)$$

where $z = \lambda_\ell^{f,s}+1$ if $i+\ell = k+s+1$ and $f = f_0$, and $z = \lambda_\ell^{f,s}$ otherwise. Since $\{\lambda_\ell^{f,s}\}$ are i.i.d. uniform random variables and also independent to f_0, we have $H(z|f_0) = H(z)$, hence the mutual information $I(\boldsymbol{Q}_\ell^{f,s}[i,\cdot]; f_0) = 0$. Therefore $H(f_0|\boldsymbol{Q}[i,\cdot]) = H(f_0)$ and the scheme is private.

Let us now compute the PIR rate. The file $\boldsymbol{X}^{f_0}$ consists of

$$(n-k)B = (n-k)(k(d-k)+k(k+1)/2)$$

symbols over $\mathbb{F}_q$. For columns $j \in \{k+1,\dots,d\}$, the user downloads k responses from each server $S_1,\dots,S_n$. Hence she gets a total of $nk(d-k)$ symbols for all these steps. For columns $j \in \{1,\dots,k\}$, the user downloads j responses from servers $S_{k-j+1},\dots,S_n$, leading to a total of $\sum_{j=1}^{k} j(n-k+j)$ symbols for those steps. Therefore, we get the following PIR rate:

$$\begin{aligned}\rho &= \frac{(n-k)\left((d-k)k+\frac{k(k+1)}{2}\right)}{(d-k)nk+\sum_{j=1}^{k} j(n-k+j)} \\ &= \frac{3(n-k)(2d-k+1)}{6dn-3nk+3n-k^2+1} \,.\end{aligned} \tag{5}$$

□

As a function of n,k,B, the PIR rate given in Theorem 1 can be written as

$$\rho = \frac{1-\frac{k}{n}}{1-\frac{k(k+1)(k-1)}{6nB}} \,. \tag{6}$$

Indeed, starting from Equation (5) we get

$$\rho = \frac{(n-k)B}{nB+\sum_{j=1}^{k} j(j-k)} = \frac{(n-k)B}{nB-\frac{k(k+1)(k-1)}{6}} \,,$$

leading to the expected expression.

Our results also provide a lower bound on the capacity of PIR schemes based on MBR codes. Notice that our achieved

rate does not depend on the number of files. Hence this rate is achievable for any number of files.

Corollary 1. *The capacity c_{MBR} of (n,k,d)-MBR-coded PIR schemes satisfies:*

$$c_{\mathrm{MBR}} \geq \frac{1-\frac{k}{n}}{1-\frac{k(k+1)(k-1)}{6nB}}\,.$$

Tightness of this bound is left as an open question.

IV. PIR RATE

Since the optimality of our construction remains undecided, we propose some numerical and asymptotic analysis of the PIR rate of our schemes.

A. Bounds on the PIR rate

Lemma 3. *Let $1 \leq k \leq d \leq n$ such that $k < n$. The PIR rate ρ of the scheme from Theorem 1 satisfies:*

$$1-\frac{k}{n} \leq \rho \leq 1-\frac{B}{nd}\,.$$

Proof. If $1 \leq j \leq k$, it is clear that $n-k+j \leq n$. Using this in Equation (5), we get

$$\rho \geq \frac{(n-k)\left((d-k)k+k(k+1)/2\right)}{n(d-k)k+n\sum_{j=1}^{k} j} = \frac{n-k}{n} = 1-\frac{k}{n}\,.$$

The right-hand-side inequality is a bit more technical to state. Using the expression of ρ given in Theorem 1, it is equivalent to prove that

$$\begin{aligned}\Delta \coloneqq\ & (nd-B)(6dn-3nk+3n-k^2+1)\\ & -3(n-k)(2d-k+1)nd\end{aligned}$$

is non-negative. A computation shows that

$$2\Delta = (k-1)[6n^2d-(2nd-2kd+k^2-k)(3n+k+1)]\,.$$

If $d=k$, then we get $2\Delta = k(k-1)(k+1)(n-(k+1)) \geq 0$ as long as $n \geq k+1$ which must hold for non-degenerated MBR codes.

If $d \geq k+1$, as it is for a non-trivial regenerating code, then we get

$$\begin{aligned}2\Delta &= (k-1)\Big(d((k-1)(4n+2k+3)+2n+2)\\ &\qquad -(k-1)(k+1)(3n+k+1)\Big)\\ &\geq (k+1)(k-1)^2(n+k+2)\\ &\qquad +2(k+1)(k-1)(n+1)\\ &\geq 0\,.\end{aligned}$$

□

B. Comparison with the multi-file PIR scheme of Dorkson and Ng

In [27] Dorkson and Ng proposed a PIR scheme over PM-MBR codes in the context of *multi-file* retrieval, *i.e.* any set of $p \geq 1$ files $\boldsymbol{X}^{f_0},\ldots,\boldsymbol{X}^{f_{p-1}}$ can be simultaneously retrieved. In the current work, retrieving p files remains possible by iterating the 1-file PIR protocol p times. Notice that this routine achieves the same PIR rate as for a 1-file PIR scheme.

In the general case, the PIR rate obtained in [27] is $\rho' = \frac{pB}{dn}$, under the additional constraint that $n = pk+d$. We notice that ρ' can be reformulated as follows:

$$\rho' = \frac{n-d}{k}\cdot\frac{B}{nd} = \frac{n-d}{n}\cdot\frac{B}{kd}\,.$$

For $k < n$ (which is the case for non-degenerate PM-MBR codes), this results into $\rho' < 1-\frac{k}{n}$, and using Lemma 3 we get $\rho' < \rho$, where ρ is the PIR rate achieved in the current work. We emphasize our improvement upon [27] with the numerical and asymptotic analyses proposed in Figure 4.

C. Comparison with the asymptotic capacities of MDS-PIR schemes

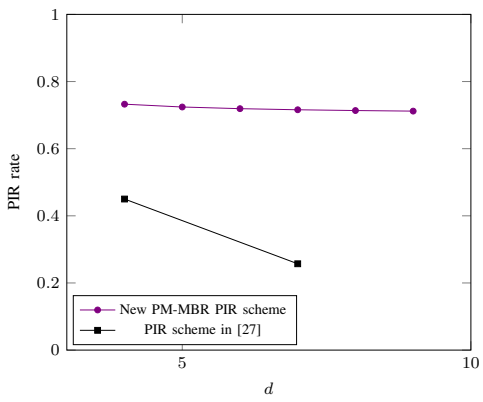
Families of (n,k,d)-MBR codes and $[n,k]$-MDS codes both allow to retrieve files by contacting only k nodes among n. MBR codes are less efficient in terms of storage, since their information rate is $B/(nd) < k/n$. The additional redundancy of MBR codes should thus lead to PIR schemes with larger PIR rate. The goal of this section is to quantify this improvement.

Our scheme is strongly linear [31], [32], hence the retrieval procedure and the PIR rate do not depend on the number of files. Furthermore, as known PIR capacities converges to the asymptotic capacity exponentially fast for $F \to \infty$, we will compare our PIR rate to the known asymptotic PIR capacities.

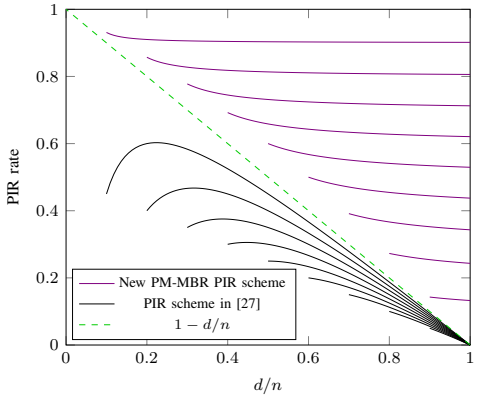
Let us outline three relevant comparisons with existing or conjectured PIR capacities.

1) The asymptotic PIR capacity of $[n,k]$-MDS-coded schemes is $1-k/n$ [9]. As we have seen in Lemma 3, our PIR scheme over (n,k,d)-MBR codes has a better PIR rate ρ. While PM-MBR codes are not MDS, this is a relevant comparison in the case where any k out of n servers must be able to recover any file. As Figure 5 illustrates, we can trade off storage space for PIR rate. Indeed, while MBR codes store more than a corresponding MDS code, they has a better PIR rates when given by the product matrix construction.
2) If we want to compare with PIR schemes over $[nd,B]$-coded data, one can model each server storing $\alpha = d$ symbols as a d-tuple of "virtual" or "sub"-servers storing one symbol each. Using this representation, those d-tuples of sub-servers collude with each other (since queries are sent to actual servers)[1]. For this reason, it is relevant to compare the PIR rate of this scheme with the (conjectured) asymptotic capacity of a PIR scheme for an $[nd,B]$

[1]Note that the assumption of full d-collusion is pessimistic since in this setting, not any d servers can collude, rather there exist disjoint sets of colluding servers that are known a priori, cf. [33].



(a) PIR rate of both schemes with $n = 10$ and $k = 3$. For fixed values of n and k and varying $d \in \{k+1, \ldots, n-1\}$, the scheme in [27] allows only a few admissible values of p, since $n = pk + d$. The larger the p, the larger the PIR rate of [27], but it remains bounded by the present scheme for every admissible value of p.



(b) PIR rate of both schemes, with an asymptotic number of nodes n. Each curve represents a distinct value of $k/n \in \{0.1, \ldots, 0.9\}$, and we plot the PIR rate versus d/n.

Fig. 4. Comparison between PIR rates of the multi-file PIR scheme in [27] and the PIR scheme in the present paper.

MDS-coded storage system allowing collusion of servers of size up to $\alpha = d$. For the very important class of *symbol-separated PIR schemes* [32] this capacity is

$$\frac{1 - \frac{B+d-1}{nd}}{1 - (\frac{B+d-1}{nd})^F}.$$

It is conjectured that the capacity converges to $1 - \frac{B+d-1}{nd}$ as the number of files grows, even without the assumption of symbol-separation [14].

3) Finally, one can compare the PIR rate to $1 - \frac{B}{nd}$, which is the asymptotic capacity of PIR schemes over $[nd, B]$-linear codes without any collusion of servers. As the specific collusion pattern [33] of our MBR PIR scheme lies somewhere between no collusion and d-collusion, it makes sense to compare to both.

Example 3. *In the PIR scheme presented in Example 2, each file contains* $(n-k)B = 27$ *symbols, and the user needs to download* $18 + 18 + 10 + 4 = 50$ *symbols. Hence, the PIR rate is* $\rho = 27/50$ *which is larger than* $1 - k/n = 1/2$*, the asymptotic PIR capacity of an* $[n, k]$ *MDS code.*

The conjectured asymptotic PIR capacity of d*-colluding* $[nd, B]$ *MDS codes is also* $1 - \frac{9+4-1}{6\times 4} = 1/2$*, and thus remains below the PIR rate of the present construction. However, the asymptotic capacity of non-colluding PIR schemes over* $[nd, B]$*-linear codes remains larger (*$1 - \frac{9}{24} = \frac{5}{8}$*), as expected.*

A comparison between the rate of the current PIR scheme with the other relevant capacity expressions is shown in Figure 5 for different values of n, k and d. We can see that the PIR rate of our scheme is larger than the asymptotic PIR capacity of an $[n, k]$ MDS code, and for a reasonably high value of d, the conjectured asymptotic PIR capacity of d-colluding $[nd, B]$ codes.

V. Conclusion

In this paper, a PIR scheme is built for the product-matrix storage systems in the MBR setting. Using the symmetry inherent to PM codes improves the download communication complexity, achieving a PIR rate larger than $1 - k/n$, *i.e.*, larger than the asymptotic PIR capacity of $[n, k]$ MDS-coded storage systems.

A main step forward in the topic would be to compute the actual capacity of MBR-coded PIR schemes. Another possible further work is to consider the setting of colluding servers. A natural idea is to adapt the constructions of Freij-Hollanti *et al.* [14], [16], [33], by replacing matrices $\lambda_\ell^{f,s}\mathbf{1}$ by concatenations of Reed-Solomon codewords. However, the extraction of the randomness — necessary to decrease the communication cost of our schemes — cannot be done as easily as in the non-colluding case, because projected random symbols interfere with themselves.

Acknowledgments

The work of J. Lavauzelle was partially funded by French ANR-15-CE39-0013-01 “Manta”.

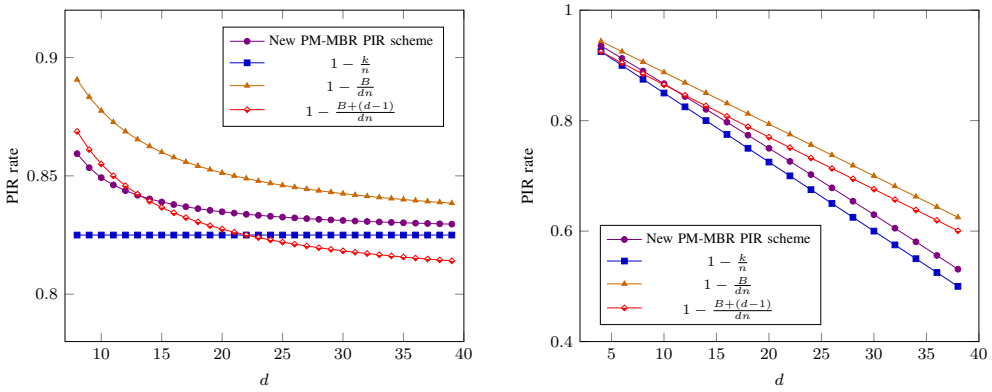
The work of R. Tajeddine and C. Hollanti was supported in part by the Academy of Finland, under grants #276031, #282938, and #303819 to C. Hollanti, and by the Technical University of Munich – Institute for Advanced Study, funded by the German Excellence Initiative and the EU 7th Framework Programme under grant agreement #291763, via a *Hans Fischer Fellowship* held by C. Hollanti.

The work of R. Freij-Hollanti was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under Grant WA3907/1-1.

References

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, “Private information retrieval,” in *IEEE Symposium on Foundations of Computer Science*, pp. 41–50, 1995.

[2] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, “Private information retrieval,” *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 965–981, 1998.



(a) PIR rate versus d when $n = 40$ and fixed $k = 7$.

(b) PIR rate ρ versus d when $n = 40$, assuming $d = 2(k-1)$.

Fig. 5. Comparison between the PIR rate of the proposed PM-MBR scheme, the asymptotic capacity $1 - k/n$ of a PIR scheme for $[n, k]$ MDS storage systems with no collusion, the asymptotic capacity $1 - B/(nd)$ of a PIR scheme for $[nd, B]$ MDS storage systems with no collusion, and the asymptotic conjectured capacity $1 - (B + d - 1)/(nd)$ of a PIR scheme for $[nd, B]$ MDS storage systems with full d-collusion. The number of nodes is chosen large enough ($n = 40$) to emphasize crossing curves.

[3] A. Beimel and Y. Ishai, "Information-theoretic private information retrieval: A unified construction," in *Automata, Languages and Programming*, pp. 912–926, Springer, 2001.

[4] A. Beimel, Y. Ishai, E. Kushilevitz, and J.-F. Raymond, "Breaking the $o(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pp. 261–270, IEEE, 2002.

[5] S. Yekhanin, "Private information retrieval," *Communications of the ACM*, vol. 53, no. 4, pp. 68–73, 2010.

[6] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Information Theory*, vol. 63, no. 7, pp. 4075–4088, 2017.

[7] H. Sun and S. A. Jafar, "The capacity of robust private information retrieval with colluding databases," *IEEE Trans. Information Theory*, vol. 64, no. 4, pp. 2361–2370, 2018.

[8] N. Shah, K. Rashmi, and K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval," in *2014 IEEE International Symposium on Information Theory*, pp. 856–860, IEEE, 2014.

[9] T. Chan, S.-W. Ho, and H. Yamamoto, "Private information retrieval for coded storage," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2842–2846, IEEE, June 2015.

[10] K. A. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Trans. Information Theory*, vol. 64, no. 3, pp. 1945–1956, 2018.

[11] A. Fazeli, A. Vardy, and E. Yaakobi, "Codes for distributed PIR with low storage overhead," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2852–2856, June 2015.

[12] S. R. Blackburn and T. Etzion, "PIR array codes with optimal pir rates," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 2658–2662, IEEE, 2017.

[13] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from mds coded data in distributed storage systems," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7081–7093, 2018.

[14] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM J. Appl. Algebra Geometry*, vol. 1, no. 1, pp. 647–664, 2017.

[15] S. Kumar, H. Lin, E. Rosnes, and A. G. i Amat, "Achieving Maximum Distance Separable Private Information Retrieval Capacity With Linear Codes," *IEEE Transactions on Information Theory*, vol. 65, pp. 4243–4273, July 2019.

[16] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, A.-L. Horlemann-Trautmann, D. Karpuk, and I. Kubjas, "t-private information retrieval schemes using transitive codes," *IEEE Transactions on Information Theory*, 2018.

[17] S. Kumar, H. Lin, E. Rosnes, and A. G. i Amat, "Local reconstruction codes: A class of MDS-PIR capacity-achieving codes," in *2018 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2018.

[18] Q. Wang and M. Skoglund, "Symmetric private information retrieval for MDS coded distributed storage," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2017.

[19] Q. Wang, H. Sun, and M. Skoglund, "Symmetric private information retrieval with mismatched coded messages and randomness," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 365–369, IEEE, 2019.

[20] A. Dimakis, P. Godfrey, Y. Wu, M. Wainright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, pp. 4539–4551, Sep. 2010.

[21] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the msr and mbr points via a product-matrix construction," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.

[22] C. Suh and K. Ramchandran, "Exact-repair mds code construction using interference alignment," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1425–1442, 2011.

[23] G. M. Kamath, N. Silberstein, N. Prakash, A. S. Rawat, V. Lalitha, O. O. Koyluoglu, P. V. Kumar, and S. Vishwanath, "Explicit MBR all-symbol locality codes," in *2013 IEEE International Symposium on Information Theory (ISIT)*, pp. 504–508, IEEE, 2013.

[24] N. Raviv, N. Silberstein, and T. Etzion, "Constructions of high-rate minimum storage regenerating codes over small fields," in *2016 IEEE International Symposium on Information Theory (ISIT)*, pp. 61–65, IEEE, 2016.

[25] M. Ye and A. Barg, "Explicit constructions of high-rate MDS array codes with optimal repair bandwidth," *IEEE Trans. Information Theory*, vol. 63, no. 4, pp. 2001–2014, 2017.

[26] M. Elyasi and S. Mohajer, "A cascade code construction for (n, k, d) distributed storage systems," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 1241–1245, IEEE, 2018.

[27] C. Dorkson and S. Ng, "Multi-message private information retrieval using product-matrix msr and mbr codes," *arXiv preprint arXiv:1808.02023*, vol. abs/1808.02023, 2018.

[28] C. Dorkson and S. Ng, "Private information retrieval using product-matrix minimum storage regenerating codes," *arXiv preprint arXiv:1805.07190*, vol. abs/1805.07190, 2018.

[29] A. Patra and N. Kashyap, "On the pir capacity of MSR codes," *arXiv preprint arXiv:1901.03809*, 2019.

[30] J. Li, D. Karpuk, and C. Hollanti, "Private information retrieval from MDS array codes with (near-) optimal repair bandwidth," *arXiv preprint arXiv:1909.10289*, 2019.

[31] L. Holzbaur, R. Freij-Hollanti, and C. Hollanti, "On the capacity of private information retrieval from coded, colluding, and adversarial servers," in *2019 IEEE Information Theory Workshop (ITW)*, 2019.

[32] L. Holzbaur, R. Freij-Hollanti, and C. Hollanti, "Towards the capacity of private information retrieval from coded and colluding servers," *arXiv preprint arXiv:1903.12552v5*, 2020.

[33] R. Tajeddine, O. W. Gnilke, D. Karpuk, R. Freij-Hollanti, C. Hollanti, and S. El Rouayheb, "Private information retrieval schemes for coded data with arbitrary collusion patterns," in *2017 IEEE International Symposium on Information Theory*, pp. 1908–1912, IEEE, 2017.